

# **A MAP BASED DECISION SUPPORT SYSTEM FOR DISTRIBUTION PLANNING**

051211

A Thesis Submitted  
In Partial Fulfillment of the Requirements  
for the Degree of  
MASTER OF TECHNOLOGY

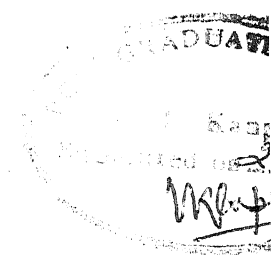
by

**YASH KANT GUPTA**

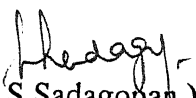
to the

**DEPARTMENT OF INDUSTRIAL AND MANAGEMENT ENGINEERING  
INDIAN INSTITUTE OF TECHNOLOGY, KANPUR  
FEBRUARY, 1995**

## CERTIFICATE



This is to certify that the work contained in the thesis entitled " A MAP BASED DECISION SUPPORT SYSTEM FOR DISTRIBUTION PLANNING" by Mr. Yash Kant Gupta (Roll No. 9311426) has been carried out under my supervision and that this work has not been submitted elsewhere for a degree.

  
(S Sadagopan)

Professor

Industrial and Management Engineering Department  
Indian Institute of Technology  
Kanpur-208 016

IME-1995-M  
GUP-M.

23 MAR 1995/IME

CENTRAL LIAISON  
F. I. T. KANPUF

Doc. No. A. 119129

IME-1995-M-GUP-MAP

## ACKNOWLEDGEMENT

Let the words that follow don't loose their power simply because it's customary to acknowledge. I had the proud privilege of association with Dr. S Sadagopan who is a store house of innovative ideas. So if this thesis appears innovative in any sense the credit is all his, any error that might have crept in is however all mine. I thank him for all his help, inspiration, encouragement- and for bringing out the best out of me.

I also had the good luck of being associated with a highly cohesive batch, so much so that we were the envy of our hostel. My sincere thanks to IME family for the help and encouragement provided to me during my M.Tech studies.

Yash Kant Gupta

## ABSTRACT

In this thesis, a decision support system has been developed to help the user to take distribution decisions. The system blends the advantages of analytical ability of management science / operations research techniques and the computational support of an interactive computer system. The basic underlying strategy in development of this DSS is the standard transportation problem with a map interface to it. The system is fully functional with a database of over 150 major cities of India including locations and relative separation. The application has been developed in Visual Basic 3.0 under Microsoft Windows 3.1 environment. A modular approach has been followed in designing the system. This enables any future modifications to the system quite easy. Throughout the design the stress has been on designing a support environment rather than in developing a simple computerized distribution system.

# CONTENTS

LIST OF FIGURES .....	vii
CHAPTER I .....	1

## INTRODUCTION

1.1 Introduction.....	1
1.2 Decision Support Systems.....	2
1.3 DSS Characteristics and Benefits .....	3
1.4 Distribution Planning .....	3
1.5 Outline of the Problem .....	3
1.6 The Graphical Revolution .....	4
1.7 Nature and Scope of Problem .....	5
1.8 Motivation.....	6
1.9 Organization of the Thesis .....	6

CHAPTER II.....	7
-----------------	---

## LITERATURE SURVEY AND PROBLEM STATEMENT

2.1 Introduction.....	7
2.2 DSS for Distribution Planning.....	10
2.3 Initiation of DSS Development .....	12
2.4 The Modeling Process.....	13
2.5 Translating the Model Into a Usable Decision-Making Aid.....	13
2.6 Application of the DSS .....	14
2.7 The System Architecture.....	14
2.8 Tools and Generators.....	15
2.9 Hardware.....	15
2.10 Development Time.....	15
2.11 Interactive / Prototyping Activities.....	20
2.12 Testing and Evaluation.....	21

CHAPTER III.....	22
------------------	----

## SYSTEM DESIGN

3.1 The System's Structure .....	22
3.2 The Dialog Sub-system .....	22
3.3 The Data Sub-system.....	23
3.4 The Model Sub-system .....	24

CHAPTER IV.....	25
-----------------	----

## SYSTEM IMPLEMENTATION

4.1 The Program Structure .....	25
4.2 Drawing a New Map.....	26
4.3 Zooming Facility.....	26
4.4 Selection of Sources and Destinations .....	26

4.5 Solving the Transportation Problem .....	27
4.6 Showing the Results.....	28
CHAPTER V .....	33
<b>CONCLUSIONS AND RECOMMENDATIONS</b>	
5.1 Conclusions .....	33
5.2 Recommendations.....	33
CHAPTER VI.....	35
<b>USER'S MANUAL</b>	
What You Need.....	35
How the System Works .....	35
The File Menu .....	36
Load Map File.....	36
Load Result File.....	36
Exit.....	36
The Mark Menu.....	36
Mark Outer .....	36
Mark Inner.....	38
The Draw Menu .....	38
Draw Directly .....	38
By Draw File.....	38
The View Menu.....	38
Full Page.....	38
Zoom In/Out.....	38
The Save Menu .....	40
Save Map File .....	40
The Data Menu.....	40
Choose Sources / Destination.....	40
Start Selection.....	40
End Selection.....	40
Distance Table .....	40
Prepare Cost Table.....	41
The Run Menu.....	41
Execute Program .....	41
Show Results .....	41
The Clear Menu.....	41
The About Menu .....	41
The Help Menu.....	41
The System's Error Messages .....	42
Nothing To Draw.....	42
.dss File Not Loaded .....	42
Source/Destination File Missing .....	42
Can't Execute.....	42
BIBLIOGRAPHY .....	43

## LIST OF FIGURES

<b>Figure</b>	<b>Title</b>	<b>Page</b>
2.1	Summary of the DSS's three sub-systems and their functions	16
2.2	The process of establishing dual and shared responsibilities between the decision support system (DSS) and the decision maker (DM)	17
2.3	The Tasks Performed by the DSS to Translate the Model into a Usable and Effective Technique	18
2.4	Interaction between the user and the decision support system in a typical application	19
4.1	Level 1 System Diagram	29
4.2	Drawing a new map	30
4.3	Data entry phase	31
4.4	Solution to the main problem	32
6.1	Basic Layout of screen	37
6.2	Basic dialog box structure	37
6.3	Digitized map displayed on screen	37
6.4	Screen showing selection of cities from a list	39
6.5	Final output displayed on user screen	39



# CHAPTER I

## INTRODUCTION

### 1.1 Introduction

Computer based technologies are being developed to improve the effectiveness of managerial decision making, especially in complex tasks. Three such technologies- decision support systems(DSS), executive information systems(EIS), expert systems(ES) are being widely used to supplement managerial decision making. They are collectively known as management support systems. Decision support systems, about fifteen years old, have already proven themselves by providing business with substantial savings in time and money[3].

All managerial activities revolve around decision making. The manager is foremost a decision maker. Organizations are filled with decision makers at various levels. Each manager is charged in fact with a certain part of an organization's decision making activities. For years, managers have considered decision making as pure art; a talent acquired over a long period of time through experience/learning by trial and error. Management was considered an art because a variety of individual styles could be used in approaching and successfully solving the same type of managerial problems in actual business practice. These styles are often based on creativity, judgment, intuition, and experience, rather than on systematic quantitative methods based on a scientific approach.

However, the environment in which the management must operate is changing. We live in information and microelectronic age, where technical advancements become a major determinant of our lifestyle. Such advancements in technology can not be made or sustained without concurrent advancement in management systems. Business and its

environment are more complex today than ever before, and the trend is towards increasing complexity. Today decision making is more complex than in past. There are three reasons for increased complexity. First, the number of alternatives are much larger today than ever before, owing to improved technology and communication systems. Second, future consequences are more difficult to predict because of increased uncertainty. Finally, the cost of making error may be very large owing to the complexity and magnitude of operations, automation, and the chain reaction that may cause in many parts of the organization.

As a result of these trends and changes it is very difficult to rely on trial and error approach to management. Managers must become more sophisticated, they must learn how to use new tools and techniques that are being developed in their field. Many of these techniques use a quantitative analysis approach; they are grouped into a discipline called management science or operations research.

## 1.2 Decision Support Systems

The concepts involved in DSS were first evolved in 1970s by Scott-Morton under the term management decision system. He defined such systems as interactive computer based systems, which help decision makers utilize data models to solve unstructured problems[22]. Another classical definition of DSS is provided by Keen and Scott-Morton [14].

Decision support systems couple the intellectual resources of individuals with the capabilities of the computer to improve the quality of decisions. It is a computer based support system for management decision makers who deal with semi-structured problems.

Having explained the definition of decision support and other systems one can say that DSS represents a point of view on the role of computers in managerial decision making process.

### 1.3 DSS Characteristics and Benefits

Following are the characteristics of DSS :

- Ability to support the solution of complex problems.
- A decision support system enables a thorough, quantitative analysis in a very short time.
- Ability to try several different strategies under different configurations, quickly and objectively.
- DSS can increase management control over expenditures and improve performance of the organization.
- Routine application of a DSS may result in considerable cost reduction, or in reducing (eliminating) the cost of wrong decisions.
- The decisions derived from DSS are more consistent and objective than decisions made intuitively.

### 1.4 Distribution Planning

Distribution planning is one area of operations research where the main thrust is on optimally planning the distribution of commodities on day-to-day basis. Such decisions are taken on a regular basis. In such problems, major emphasis is put on cost reduction and timeliness of decision making cycle. Because of the frequency of such decisions, they may affect the firm's profits adversely if not taken intelligently.

### 1.5 Outline of the Problem

A real life problem of distribution of commodities like oil, food grains, fertilizers, coal, steel etc. consists of optimally distributing the commodities from a set of originating points to a set of distribution points. The general system of distribution is as follows: as soon as the request from any demand point comes to the source, it

releases the amount ordered depending upon the availability. The commodity is transferred from the supply point to the consumption point with the help of tankers, pipelines, trucks etc. The firm would like to follow a system which can decide how much quantity shall be shipped from one source to different destinations, so that cost of transport, and delivery time is minimized. The work in this thesis aims at developing a software system which derives an operationally feasible movement plan for the distribution planning. The major feature of the system is its graphical user interface, which allows the user to run the application in an interactive manner. The system works on the model of standard transportation problem[18, 24] with generalized constraints on supply and demand. The user can interactively choose the sources and destinations displayed on the monitor which are simultaneously displayed on the map. The distances are automatically looked from a common database. The system solves the problem and displays the results on a map.

## 1.6 The Graphical Revolution

The ongoing trend in PC computing environment is toward more graphical, visual user interfaces. The typical computer user is now accustomed to software applications that feature drop-down menus, a variety of colors and fonts, and multiple windows.

Rather than typing archaic commands from the keyboard, today's users regularly move the mouse pointer onto an appropriate icon or menu option, then click the mouse button to activate a command or initiate a program. In current PC market, Microsoft Windows is the epitome of the graphical user interface. Windows applications have a consistent look and feel, featuring icons, mouse support, drop-down menus, and resizable windows.

Visual computing works because it is intuitive. People are visually oriented and commonly interact with their world by responding to visual clues. The more visual their

computing environment, the more natural it feels. All graphical user interfaces make use of graphics on a bit mapped video display. Graphics provide better utilization of screen real estate, a visually rich environment for conveying information, and a possibility of a WYSIWYG (what you see is what you get) video display of graphics and formatted text prepared for printed document.

In earlier days, the video display was used solely to echo text that the user typed using the keyboard. In graphical user environment the video display itself is a source of user input. The video display shows various graphical objects in the form of icons and input devices such as buttons and scroll bars. Using the keyboard (or, more directly, a pointing device such as a mouse), the user can directly manipulate these objects on the screen. Graphics objects can be dragged, buttons can be pushed, and the scroll bars can be scrolled. The interaction between the user and the program thus become more intimate. Rather than one-way cycle of information from the keyboard to the program to the video display, the user directly interacts with the objects on the display. The application developed under this thesis has a map interface built into it. Graphical user interface in the form of a map gives a visual clue to the user regarding the nature and type of problem, also the user feels more tuned to the problem environment.

The system also behaves in some ways like a GIS. Geographical information system (GIS), are a powerful tools for collecting, storing, retrieving at will, transforming and displaying spatial data from a real world for a particular set of purposes[2].

### **1.7 Nature and Scope of Problem**

The problem under consideration i.e., standard transportation problem[18, 24], is well known in the field of operations research. The problem structure is as follows :

- The objective function is to minimize the total cost of transportation of a commodity from different originating points to the various consumption points.

- Generalized constraints on source capacity and destination requirements are applicable.

The idea is to solve the problem by giving a map interface to it i.e., input is taken from, and output is displayed on, a map, on the computer screen. Not only the system provides a set of maps, the user can himself create or digitize maps with the help of widely used selection device called mouse.

## 1.8 Motivation

The transportation problem solution strategy adopted in this thesis is somewhat different from others. Here the aim is to solve it in such a manner which is visually appealing to the user. The problem environment is well known, but the solution methodology is focused on to make the application more user friendly and reusable. We have tried to give a map interface to the problem. With this interface there will not be a need to reenter distance data as it will be extracted from a common database. The approach adopted here combines GUI, database, GIS, and modelling into one consolidated system.

## 1.9 Organization of the Thesis

In this thesis an attempt is made to develop a system, which will be helpful to the user in managing distribution problem without directly handling large amount of data. Chapter I gives a broad view of the emerging technologies in the field of decision sciences. Some idea about the present user friendly computing environment is also given. Chapter II consists of a literature survey on the topic of thesis and shows all the broad steps needed to generate an application from scratch. Chapter III and IV basically deals with the system design and implementation schemes. Chapter V is about the conclusions and recommendations. System operating instructions in the form of a concise user manual are given in Chapter VI.

## CHAPTER II

# LITERATURE SURVEY AND PROBLEM STATEMENT

### 2.1 Introduction

Transportation problems are generally concerned with the distribution of a certain product from several sources to numerous locations at minimum cost. Linear programming formulation of such problems is given by Phillips and Ravindran[18], Hillier and Lieberman[12], Taha[24] any many others. In a survey of American industries[6], linear programming came out as the most often used technique(74%) among all the optimization methods. It has been reported in Science News [23] that about one-fourth of computer time spent on scientific computations in recent years has been devoted to solving linear programming problem and their many variants. The oil companies are among the foremost users of very large linear programming models, using them in petroleum refining, distribution, and transportation. There are many books on this topic but two excellent books Gass[7] and Salkin and Saha [20], which are devoted solely to linear programming application in diverse areas like defense, industry, retail business, agriculture, education and environment. many of the application also contain a discussion of the experience in using linear programming model in practice.

Discussion of various computer solutions to large linear programming problems in industry and computational features of linear programming software are discussed in Driebeek[5], Daellenbach and Bell[4], and Bradley, Hax, and Magnanti[1]. An excellent bibliography on linear programming applications is available in Gass[7], which contains a list of references arranged by area (e.g., industry, military, transportation, production, agriculture). In the area of industrial application the references have been further

categorized by industry (e.g., chemical, petroleum, iron and steel, paper, airline, railroad). Some other references are in a survey conducted by Gray and Cullinan-James[9]. Some recent application of linear programming are given in Interfaces, AIIE Transactions, Decision Sciences, European Journal of Operational Research, Management Science, Operation Research, Operation Research (UK), Naval Research Logistics Quarterly, and Opsearch (India).

Solution of transportation model by simplex method is inefficient because it does not exploit the special structure of the problem. An important feature of the standard transportation problem is that, it can be expressed in the form of a table, which displays the values of all the data coefficients associated with the problem. In fact, the constraints and the objective function can be read off directly from the table. Because of the special structure of the transportation matrix, it is very easy to find out the initial basic feasible solution to start the simplex method without the use of artificial variables. A number of methods described in Phillips and Ravindran[18]. To name a few North-West corner rule, The Least Cost rule, Vogel's approximation method are there. Optimality can be checked using modified distribution method (or better known as u-v method) given in the same text. Book also describes a number of methods to improve the basic feasible solution. A number of labeling schemes to find out theta-loop are given in Murty[17]. Other methods for labeling schemes based on networks are discussed by Johnson[13], Glover, Karney, and Klingman[8] developed the application of Johnson's tree labeling method.

Sarma[21] presents the reduced matrix method to solve the standard transportation problem. It consist of systematic transformation of the original cost matrix by a sequence of operations involving only addition and/or subtraction of positive constants to all the elements of a row or a column until the feasible solution is found by making positive assignments to the zero containing cells of the transformed matrix. The transformed matrix contains only non-negative elements and is called the



reduced matrix. The paper gives a comparison of many algorithms with his method to solve the standard transportation problem and computing experience on small scale problems indicates that his method is faster than most of the other conventional methods.

Gupta and Lenin[11] have given a software system which gives an operationally feasible plan for fertilizer movement. The system generates a model of its own and the user is allowed to do modification and updation in supply and demand constraints. The system incorporates constraints normally existing in a transportation and distribution situation. Greenwood and Moore[10] have given a decision support system in which they have described the basic steps involved in planning, designing, implementing, and testing the system as a one integrated unit. They have given a realistic view of all the problems normally occurring in the development of the big system. Keen[15] has mainly stressed on the steps needed to convert an idea into a decision making tool. Turban[25] has given an in-depth studies and the implementation procedure adopted for a number of cases discussed . He has mainly emphasized on the modularisation of big software so that in future the system can be changed easily. Greenwood and Moore[10] mainly emphasized on the fact that the system should be designed in such a way that it will give answers to the user's what if kind of situations. Medsker[16] has given a number of hints to ensure the successful implementation of any DSS. He has emphasized on some features and capabilities of a good DSS. This includes the idea of adaptive filtering in which the system gives only that much of information which is required by a particular issue and information overload is avoided. Secondly he points out on the fact that, the information once analyzed must be available for other steps too and it need not be analyzed again and again. Thirdly, he says that the user should have access to a variety of the capabilities including the ability to combine, weigh, alter, and otherwise put information into usable, convenient forms.

In this thesis the initial solution was calculated using the North-West Corner rule. The method does not give the solution which is near to the optimal but this has been used considering the easiness in coding so that more time can be devoted towards developing the interface. The u-v method used in checking the optimality of solution. But data structures have been used to make the program fast.

## 2.2 DSS for Distribution Planning

Now we will describe the design, construction, and implementation of a comprehensive computer-based decision support system that serves as an interactive decision making environment for distribution planning.

This DSS is developed to plan the distribution system of the firm.. The system is developed to enhance the firm's ability to deal effectively with distribution problems at a detailed and sophisticated level without directly handling large amount of data. The DSS is intended to support the planning, decision making, and policy setting processes by providing a way to readily analyze the effect of changes in the variables and parameters that impact the distribution of commodities. The targeted users of this system are middle level administrators directly involved in decision making activities of the firm.

Distribution of commodities forms the inseparable part of the firm's day-to-day routine work. With ever increasing demand of products and the number of supply and consumption points, the firm's administrators task of making distribution policy decisions is becoming more critical and more difficult. Hence, handling of enormous amount of data can't be done effectively and efficiently without the help of computer. As a result of this managers are bound to take help of fast computing aids. With ever increasing cost of transportation, it is necessary for the company to plan the distribution activities in a more judicious way.

A major concern for establishing a plan for distribution is the distance covered to deliver the amount from a particular source to a particular consumption point. Here we are assuming that the vehicle is taking the shortest possible route to reach the destination and no transshipment is there. Overall, the total cost of transportation is a function of source capacity, destination requirement, and the cost of transportation from a source to a destination. In case of any shortages ( i.e., when the sum of source capacities is less than the sum of requirements of all destinations ), even though the demand can not be met, one may want to find out the least cost shipping schedule which will supply as much as possible to the markets. And, this way the shortage is equally experienced by each destination.

When total supply exceeds the total demand, a dummy market is created to absorb the excess supply available at the warehouse. The unit cost of shipping from the dummy source to any destination is taken as zero since there is no actual transfer of goods taken place.

The development of DSS structure is further complicated by decisions made in an uncertain environment confounded by changing demand and supply conditions. Tackling such problems manually is not possible without the help of computerized methods, which helps decision maker to arrive at a concrete solution at a click of button. The distribution planning model also requires assimilation of large amount information, the application of correct model, and the consideration of large number of interrelated variables.

To address these problems and thereby enhance planning decision making, the quick information retrieval and modification analysis, a decision support system with an embedded optimization programming model was developed.

### 2.3 Initiation of DSS Development

No formal need assessment or cost benefit analysis was conducted prior to the initiation of DSS. Rather, the DSS was created as a part of a natural process described as follows.

It has been recognized that the age of computerized offices and organizations has come where more and more decisions, on a routine basis need to be taken quickly and easily. Ease of use of any software mainly lies in its ability to interact with the user. With the advent of Microsoft Windows compatible software, the degree of user interface with the system has increased manifolds. With the help of visual interactive aids, users are able to learn newer techniques easily and quickly. Gone are the days of keyboard interfaces with practically no help from the software side. With ever refining graphical user interface (GUI), learning time is not a bottleneck in the experimentation process.

The concept for the development of the DSS evolved from several studies performed on some existing software available for the same purpose. As a result of these studies it became apparent that many of the available software do not provide much support to the user to make his task easy. Also the process of taking decisions was becoming both increasingly critical and difficult.

The first step taken by us was to analyze the problem environment and the solution methodologies available for modeling. One point was very clear that the problem does not involve optimization of many functions hence, single criteria decision making has to be done. The DSS development in this thesis began with a set of preconceived plans for its development. To name a few were the source database, destination database, distance database, and a set of maps.

## 2.4 The Modeling Process

The least cost shipping schedule is represented as a single criteria, standard transportation programming problem. Following is the mathematical model of the problem.

$$\min Z = \sum_{i=1}^{i=n} \sum_{j=1}^{j=m} C_{ij} X_{ij}$$

subject to

$$\sum_{j=1}^{j=m} X_{ij} = a_i \text{ for } i = 1 \text{ to } n$$

$$\sum_{i=1}^{i=n} X_{ij} = b_j \text{ for } j = 1 \text{ to } m$$

$$X_{ij} \geq 0 \quad \forall i, j$$

where

$X_{ij}$  : amount shipped from source  $i$  to destination  $j$ ,

$C_{ij}$  : unit cost of transportation,

$a_i$  : capacity of source  $i$ ,

$b_j$  : requirement of destination  $j$ ,

$i$  : number of sources,

$j$  : number of destination.

## 2.5 Translating the Model Into a Usable Decision-Making Aid

The core of the DSS is the transportation model as shown in Figure 2.3. However, the emphasis of the DSS is on translating the transportation model into a usable and effective technique for the administration. The DSS handles the arduous task associated with model. This include formulation, calculating parameter values, establishing targets, specify the program structure, and analysing the solution. The DSS performs certain supporting tasks like : processing and storing data, interfacing between the user and the system, providing displays, generating reports, integrating functional routines, and controlling the overall process.

Once the necessary data is available to the system, control of program shifts to the system. DSS automatically supplies the data into the program model, optimizes it and then sends it for report generation. Once this has been completed, control is then returned to the user for the purpose of exploring and testing alternatives. The schematic of this process is shown in Figure 2.4.

## 2.6 Application of the DSS

The specific steps involved in the typical application of the DSS are shown in Figure 2.2. The diagram at the left of the figure indicates that user has the choice of starting the application, providing data, creating maps, creating and storing reports, and analysing data. The user can loop through the process to solve different problems. The responsibilities of providing and analysing data is on the user.

## 2.7 The System Architecture

The architecture of the DSS is defined in terms of three highly integrated sub-systems : dialog, data, and models. These three sub-systems and their functions are summarized in Figure 2.1. the dialog sub-system controls most of the operations and provides the interface between the user and the system. The memory and information handling functions are provided by the data sub-system. The analytic function for calculating parameter values and solving the transportation model is provided by the model sub-system.

The dialog sub-system is most important because it provides the interface between the user and the system. It is linked to the model sub-system to control the models operation and data sources. It is linked to the data sub-system to provide the user with displays, reports, and memory aids and to store the user input values in the data fields. The dialog sub-system comprises of various features.

- It is primary link between the user and the system.
- It controls and manages the user session through menus, displays and control mechanisms that initiate operations when certain condition exist.
- It assign tasks and operations to various sub-programs

Other basic function which it performs are- data control, problem formulation, problem solution, and report generation.

The data sub-system, through its memory function, integrates the dialog and model sub-system. It organizes, processes and stores information as well as manipulates and converts data for use in the different components of the DSS. The data subsystem supplies input data to the models, provides work space for intermediate calculations, stores output reports and maintains displays and memory aids for the user.

The model sub-system, which provides the analytical portion to the DSS has many functions which includes calculation of the values of decision variables for all iterations , checking optimality of solution, and improving solution after each iteration.

## **2.8 Tools and Generators**

The application is generated using :

- Microsoft Windows 3.1 environment.
- Visual Basic 3.0 is used as the basic programming language. All utilities available in the program are made using same programming language.

## **2.9 Hardware**

The system is implemented on a IBM compatible, Intel processor 80386 DX based machine with Microsoft Windows 3.1 and DOS 6.0 as the disk operating system. For optimum performance of the application, the system should have at least 4MB of RAM, and 5MB of free hard disk space. DOS version 4.x or higher is acceptable.

## **2.10 Development Time**

Most of the DSS was designed, programmed, tested within 4 months time. Although no records were kept on the time spent on this process, the best estimate is approximately 750 man hours. The Visual Basic programming language environment is very new to us, hence, a good amount of time has been devoted to get hold of the language part. The task of integrating all the components and getting all the sub-

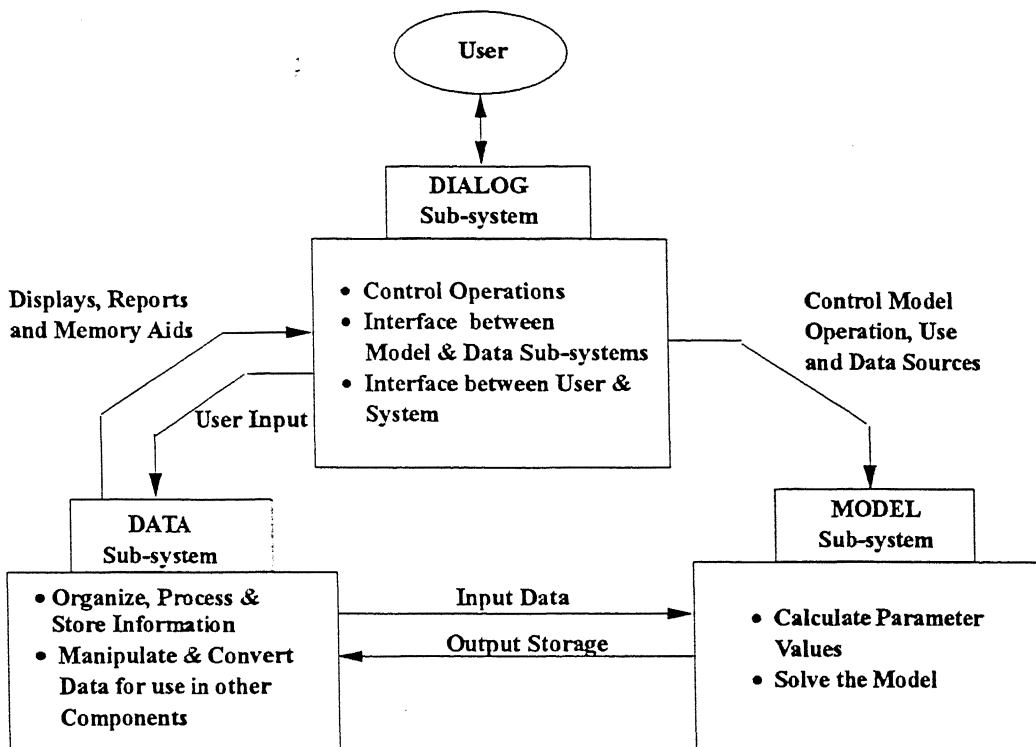


Fig. 2.1 Summary of the DSS's three sub-systems and their functions[25]



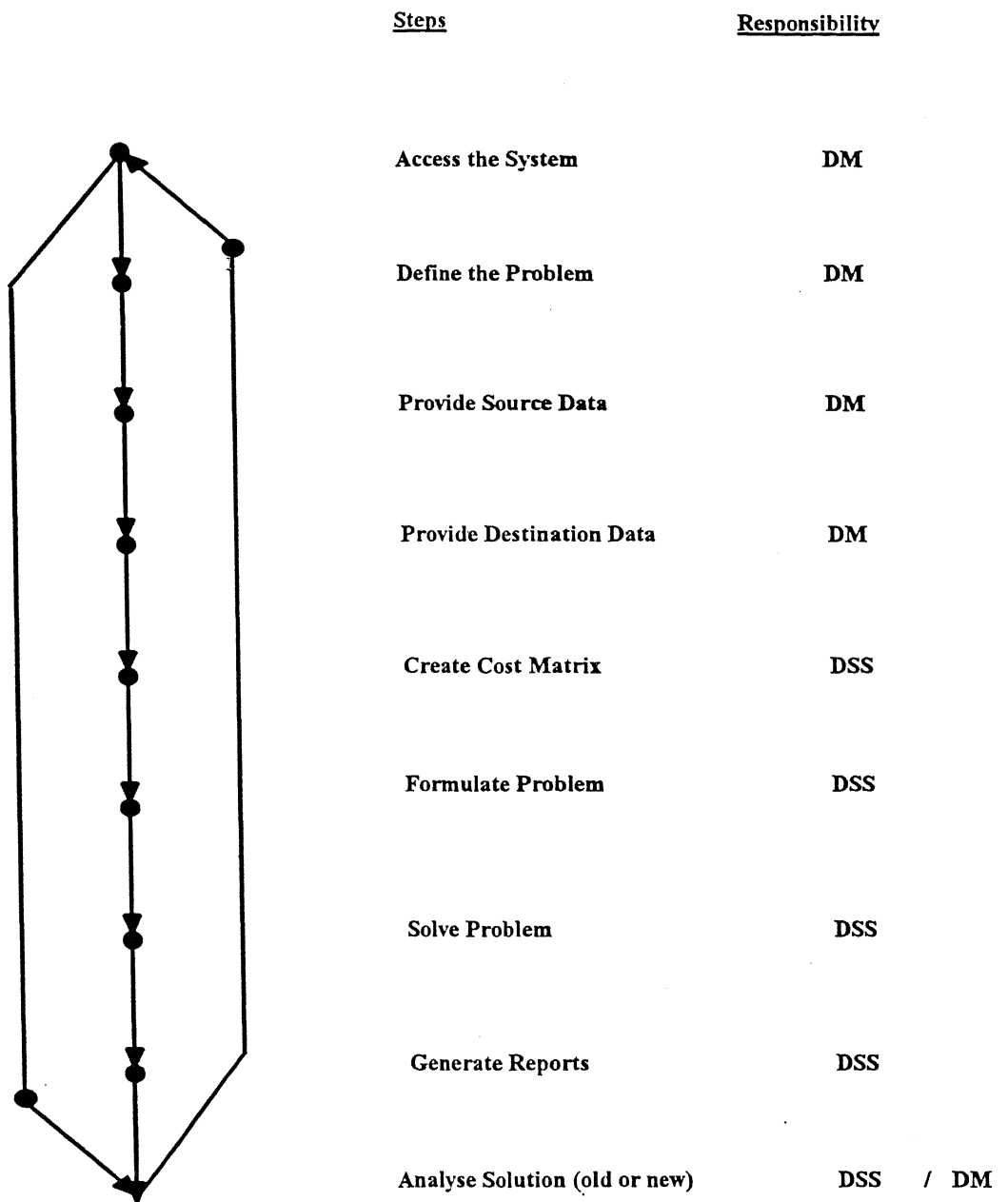


Fig. 2.2 The process of establishing dual and shared responsibilities between the decision support system (DSS) and the decision maker (DM)[25]

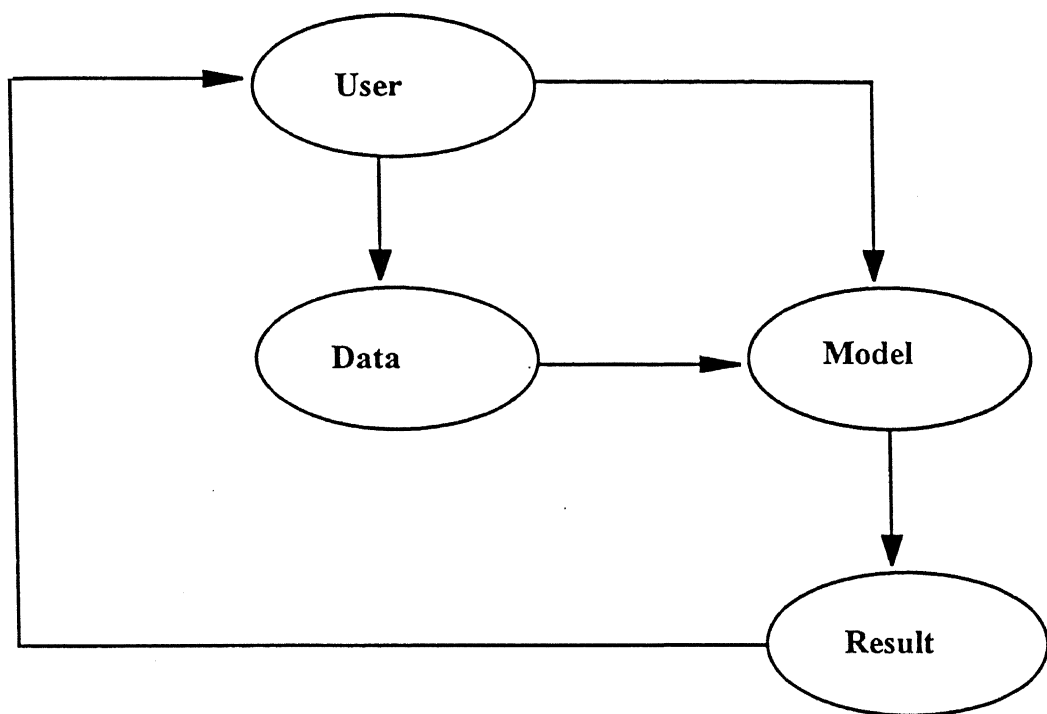


Fig. 2.4

Interaction between the user and the decision support system in a typical application[25]

systems "talking" to one another accounted for large portion of the development time(extensive programming was required ). The interface and integration added greatly to the complexity of the system.

### **2.11 Interactive / Prototyping Activities**

The interface and models within the system were tested as the system was developed. The DSS modular structure greatly facilitated to the development process, because small, manageable portions of the system could be developed and integrated into the overall system. The system's modularity also makes it conducive to modification, extension and updating at a later stage.

The philosophy used in the development process was to have the large complex system evolve from a series of small operable and proven components. The system's development was iterative in that quite often the components had to be modified after they were integrated into the system. Modifications were necessary because all the requirements of the system could not be foreseen initially as the development process progressed. The modular structure facilitated this process because the components that had to be modified could often be changed and tested apart from the overall system.

The model developed to find out the optimal solution to the generalized transportation problem was tested separately before being integrated into the system. All the model, data, and dialog routines were then integrated into the overall system, once they were found to be operating correctly and were satisfactory with the user. The links between the various programs and the interactive interface between the user and the user underwent an evolutionary development as well.

## 2.12 Testing and Evaluation

Some review sessions were conducted as the system became operational. Some examples were taken from different text books[18, 24] and checked so that system's features could be examined as possible. That testing was an on-line "walk through" of typical applications of the system. Many discrepancies found during the testing session were later removed.

CONTROL LIBRARY  
I. T., KANPUR

Acc. No. **A. 119129**

## CHAPTER III

# SYSTEM DESIGN

### 3.1 The System's Structure

The decision support system presented in this thesis consists of the following components. All the parts are complementary to each other and each component has its own importance.

- The Dialog Sub-system
- The Model Sub-system
- The Data Sub-system

### 3.2 The Dialog Sub-system

As the name suggests the dialog sub-system works as the main interface between the user and the system. This sub-system is designed and developed using Visual Basic 3.0. The programming environment has following advantages. First of all it provides window facility to make the application similar to any other application running under Windows, secondly, this helps in making the application reusable so that it can be used in association with other applications also, thirdly and most importantly, it is a strong medium for creating visual look to the application. The dialog sub-system has two other important functions to perform, firstly, it controls the flow of operations within the system and secondly, it works as a link between other components of DSS. In the present context the dialog sub-system performs the following tasks.

- It prompts the user to select options from the menu to perform different activities e.g., to load a map file, digitize map, display map either from the draw file or directly, display results, etc.

- It prevents crash down of the system by ensuring that the user always follow the correct and logical sequence of operations. Each time when the user deflects from the correct sequence, the dialog sub-system displays an error message and directs the user about the correct sequence e.g., if the user tries to execute the program without providing the system one or more data file(source file, destination file and/or cost file), the system displays an error message about the file which is not available.
- When the data required is available, the dialog sub-system automatically transfers the control to other module of the DSS to perform the next set of activities e.g., as soon as, all the data files are available, the dialog sub-system passes an internal message to model sub-system to solve the model.
- It also generates displays of various kind, may it be of information to the user(in the form of an error message) or a question(in the form of path of a file required) or in case their is any error due to less memory available the it warns the user to do some kind of house-keeping job to increase the memory for system's use. When the user do not save the require file, the system asks the user to save the necessary file.

### 3.3 The Data Sub-system

The second important part of the DSS is the data sub-system. This sub-system works as an interpreter for the model sub-system because the data which is available after each step of operation has to be converted to the form compatible with the model sub-system so that the sub-system does not have any difficulty in processing or using the data supplied. It helps in maintaining the integrity of the whole system. The data sub-system works internally and it does not have any kind of interaction with the user. The working of data sub-system is summarized in the following points.

- It organizes, processes, and stores the information. Its working can be explained in the following example. Suppose user has selected a list of sources and destinations.

Now it is the work of data sub-system to store the data in such a way which allows efficient retrieval of data when the data is required by the model to create a cost data file. This cost data file is needed by the main program as the basic input to solve the transportation problem.

### 3.4 Model Sub-system

Undoubtedly this is the most important component of the DSS. This sub-system is responsible for the proper working of the DSS. Once necessary input is available from the other two sub-systems, the model sub-system moves through the various routines of the main program to calculate a series of values which it uses for its intermediate calculations and supplies them to the next iteration. In the last when the optimization criteria is found to be true, it sends a signal to the dialog sub-system to tell the user that the DSS has found out a solution to the problem and user may analyze it. The sub-system sends the information to the data sub-system for storage. Following points with the help of examples will clear the concept of model sub-system.

- It calculates values of decision variables at each step and checks for the optimality of the solution. If the solution is found to be optimal then it sends the data for the data sub-system for storage, otherwise, control of program shifts to the next iteration. This sub-system also generates a series of flags to test the integrity of the system in each step.
- It integrates the functional routines and creates a unison among them. Although the system consists of a series of functional routines, but, as we have explained earlier that all the three sub-systems are complementary to each other so are the functional routines e.g., consider solving the main problem of optimization, the flow of program switches between a number of functional subroutines. To name a few are the function to check the optimality, subroutines to find out initial basic feasible solution, to find out theta loop, and to find out the improved solution etc.

## CHAPTER IV

# SYSTEM IMPLEMENTATION

The system was implemented on the line of structure which was formed during system design stage. All the steps taken in the previous chapter for designing purpose have been transformed in the form of flow charts to make the logical sequence of program clear.

### 4.1 The Program Structure

The system is made up of small manageable modules. Programming environment allows each basic component like, button, menu item etc. to be treated as an object. Each object has its own characteristics. Depending upon the type of message passed to the component, it behaves in the predefined way. The whole programming environment is event driven. So, whenever a message is passed to a particular object e.g., when button is pressed or clicked certain event occurs. Events are associated with some function or duty to perform. Each time an event happens it executes the associated function.

Visual Basic has the following structure of programs made in it.

- Window, a program may have a number of windows.
- Controls like button, menu etc.
- Modules, where no window is associated with it. It performs only computational work.

Following is the description of the application developed in terms of the basic entities of the system. Overall program structure is shown in Figure 4.1. Various components of the program are as follows.



## 4.2 Drawing a New Map

The system asks the user to click points located on the periphery of the loaded bit-mapped(or any other format) image of the map to be digitized. System senses the location of the point on the screen and returns it to the data sub-system. Data sub-system takes care of all such points clicked. When all points have been clicked and the user selects the option to draw a map, the system connects all the outer points of the map using polyline function and all the inner point are displayed individually. The data sub-system stores the map data in a file with an extension `.dss`(i.e., `<filename>.dss`). The file is sequential in nature and it contains data in the following way.

- Number of points on the outer boundary.
- X location of the point, Y location of the point.
- All inner points will be stored as follows : X location, Y location, name of the city.

The schematic of the procedure described above is given in Figure 4.2.

## 4.3 Zooming Facility

As and when the user tries to scale-up or scale-down the map, the system reads all the points from the `.dss` file and multiplies them with necessary multiplying factor. Because the points are connected using polyline function, map drawing on the screen is very fast and system takes very little time to update the locations of all the points.

## 4.4 Selection of Sources and Destinations

System flow chart is shown in Figure 4.3. The system presents an interactive way of selection, where all the sources and destinations are selected from the list displayed on the screen. The dialog sub-system and the data sub-system are the two main components involved in this. As the user selects any city from the list, a procedure associated with the selection gets executed. The data sub-system takes the control of

program and stores the information in an array (either capacity or requirement). When the selection is over, the data get stored in a file in the following format.

- Number representing the count of a particular city, (X, Y) coordinate of the city on the map, name of the city.

When all the sources and destinations are selected, system becomes ready to prepare cost matrix table. The cost matrix table is a file which takes input from the source file, destination file, distance database, and the arrays containing values of capacities of different sources and requirements of different destinations. The data is stored in the following form.

- Number of sources.
- Number of destinations.
- Capacity of each source.
- Requirement of each destination.
- Cost of transportation from different sources to different destinations.

#### 4.5 Solving the Transportation Problem

The flow of information is depicted in the flow charts. The system takes input from the data file i.e., cost matrix file created in the previous step. The first step is to find out the initial basic feasible solution using the North-West Corner rule. After this optimality of the solution is checked. As we know that in a transportation table number of basic variables is  $(m+n-1)$ , where,  $m$  and  $n$  are the number of sources and destinations respectively. There is no point in handling total  $(m \times n)$  number of decision variables at a time, because the number of variables needed to calculate  $u_i$ s and  $v_j$ s are only  $(m+n-1)$ . So we have used data structures to handle only the necessary variables. When the solution is optimal, it saves the solution in a file. In case a solution in any iteration is found to be non-optimal, the system improves the solution and again checks

the optimality. This cycle continues until solution is optimal. Once the solution is optimal the data sub-system stores the solution in the following format : source reference number, destination reference number, quantity shipped. The flow of program is clear from Figure 4.4.

#### 4.6 Showing Results

This procedure takes input from the answer file stored during program execution. The first entry shows the source reference number. It searches through the source file for the matching source and fetches its location coordinates on the map. Now the system draws a line between the source and the destination, and prints the quantity transferred at the middle of the line.

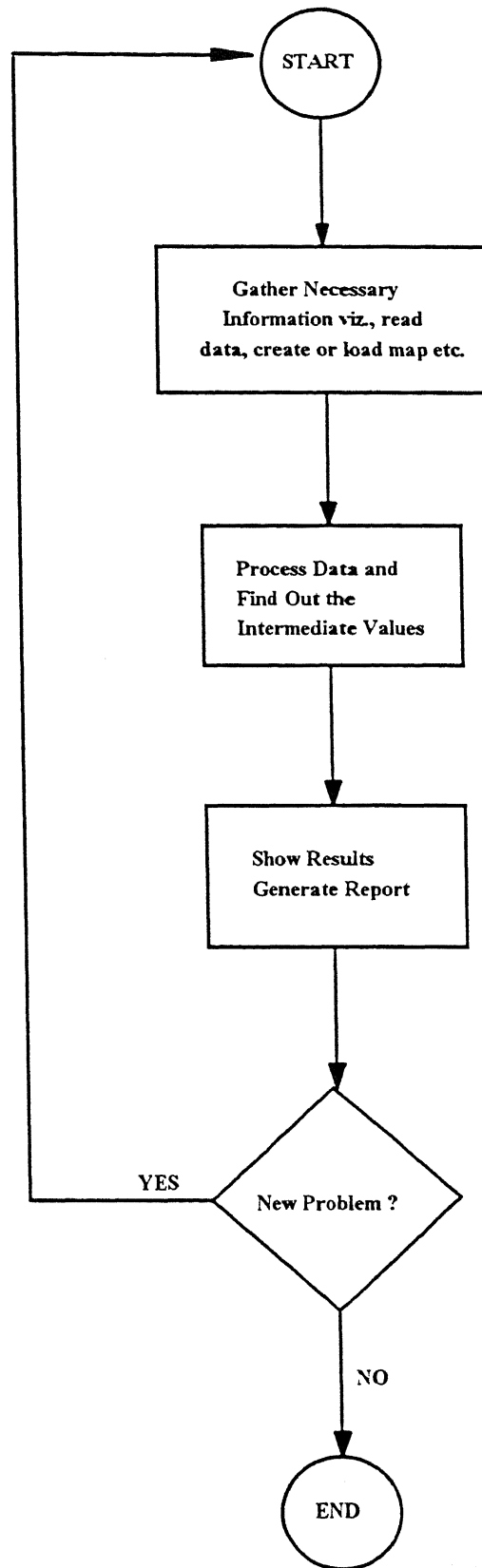


Fig. 4.1

Level 1 System Diagram

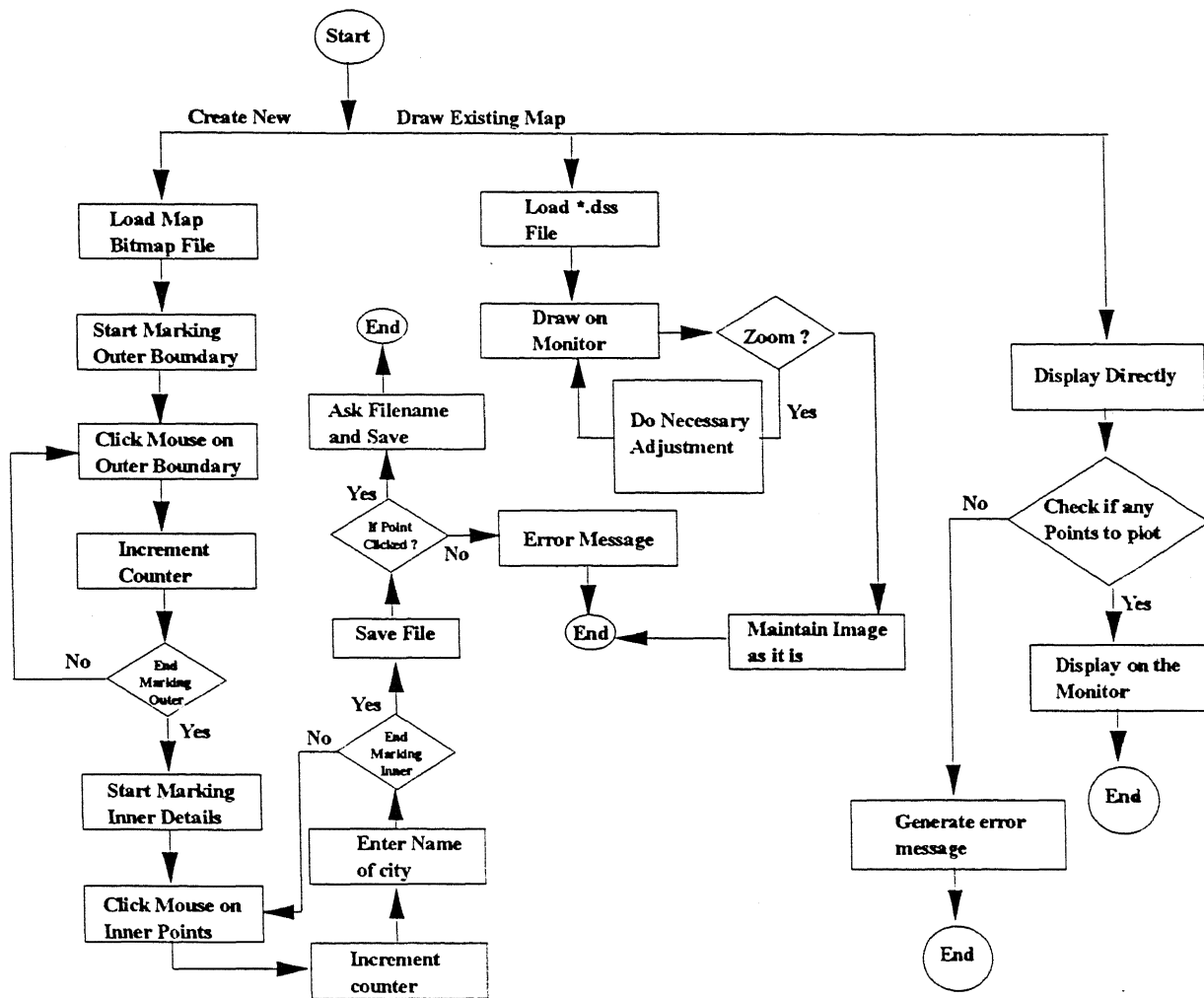


Fig. 4.2

Drawing a new map

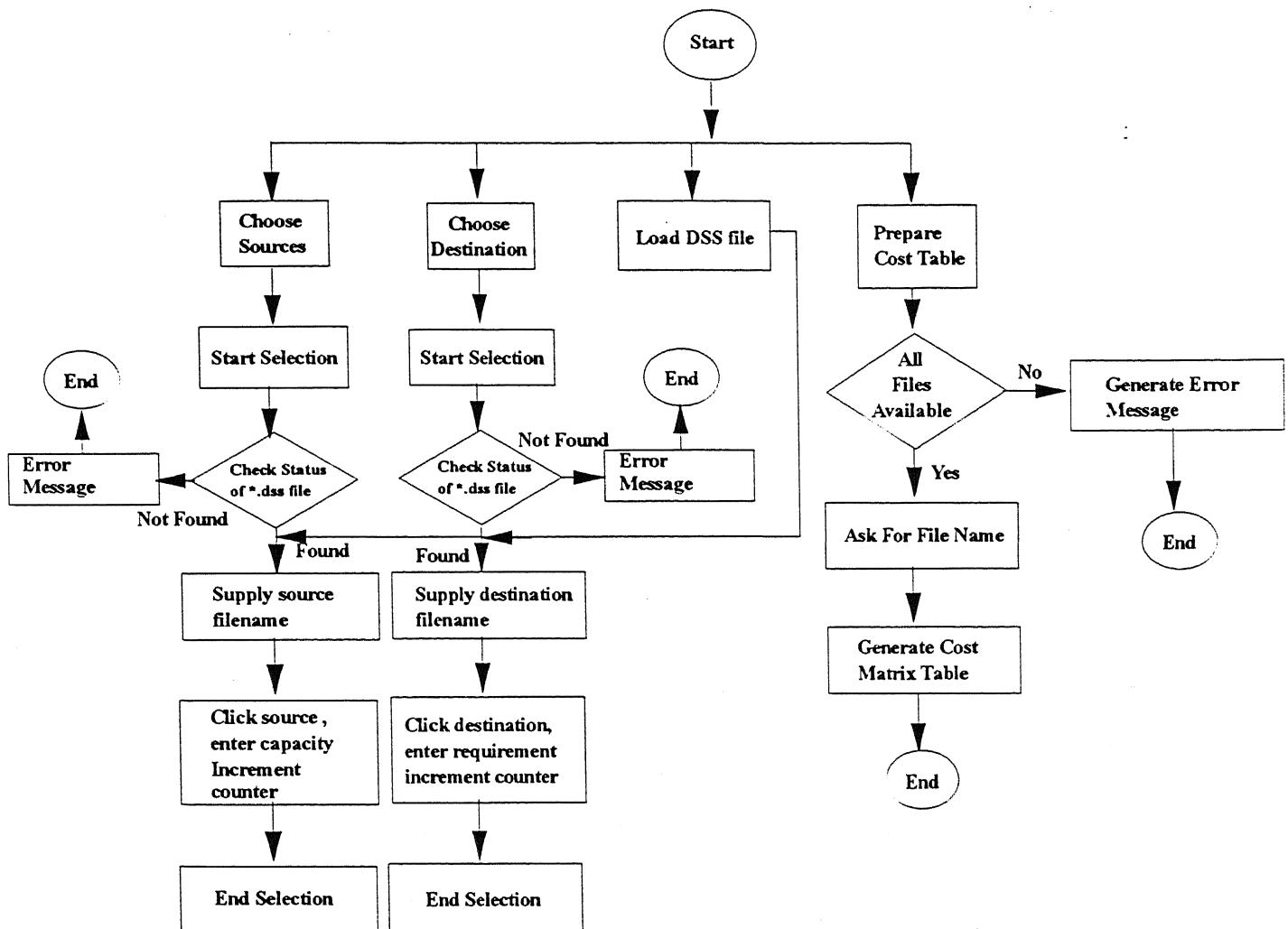


Fig. 4.3

Data entry phase

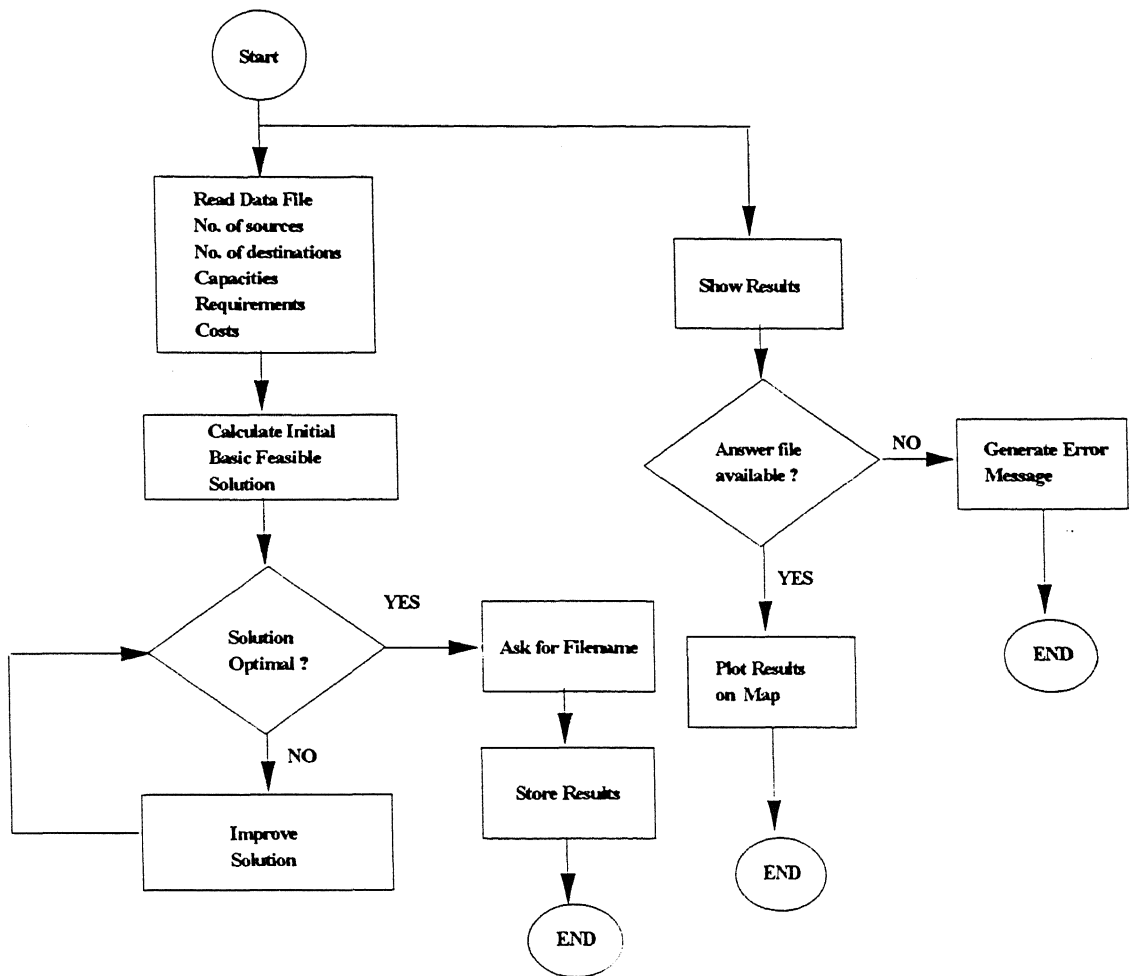


Fig. 4.4

Solution to the main problem

## CHAPTER V

# CONCLUSIONS AND RECOMMENDATIONS

### 5.1 Conclusions

The problem discussed in this thesis is well known and solvable by different algorithms. But the point is to use the available techniques to make it user friendly with an interface that is natural, intuitive and specific to the problem, viz., a map interface. At the same time efforts have been put in to implement it into Windows environment so that we can move towards frontier technologies. The application has been tested on a sufficiently large number of transportation problems to check the validity of the program. The user interface has its visual appeal and the user feels comfortable with the problem environment. The system can take various constraints on the demand as well as on supply and calculates the solution in a changing environment. Modular approach has been used right from the design stage upto the implementation stage. Hence, the program can be modified easily to incorporate the changing needs. The map interface has been tested and found effective and intuitive.

### 5.2 Recommendations

Though highest efforts have been put in to make the decision support system useful and user friendly, but following improvements if included, will make the application more sophisticated.

- The system can only digitize scanned images of A4 size only. The image should be complete in itself.
- No provision has been given to depict the actual path of movement, though path length used to solve the model is the actual one.
- Images can not be magnified or condensed horizontally and vertically by stretching them in one direction only.



- While we have concentrated on the distribution problems in this thesis, several other network flow problems can be implemented on the common interface.
- The present system works on a database of over 150 major cities of India including locations and relative separation<sup>1</sup>, but can be extended to incorporate any number of cities.

## CHAPTER VI

# USER'S MANUAL

This chapter provides step-by-step instructions for using the DSS and associated files on PC. The instructions given here assume that the user is familiar with Microsoft Windows, including standard mouse techniques such as clicking and dragging. For more information about mouse techniques, working with disks or making copies of documents and disks, Microsoft Windows user's guide can be used for reference.

### What You Need

To install and use decision support system (DSS), you need the following equipment and hardware.

- any IBM-PC or equivalent with an 80386 microprocessor or higher.
- at least 2MB of RAM, but 4MB or more provides optimum performance.
- Windows system software version 3.0, or higher, and MS-DOS version 3.1, already installed on your PC's hard disk drive. Windows version 3.1 and MS-DOS version 5.0 is recommended.
- a hard drive external or internal.

### How the System Works

The text describes the step-by-step procedure to run the program effectively. All options provided in the application are described in a sequential fashion. The main menu provides choices like File, Mark, Draw, View, Data, Run, About, Save, Help and Clear, each of which is described in the text to follow.

## **The File Menu**

The file pull down menu offers various choices for loading existing files either for map digitization or for result analysis. The keyboard short-cut for this menu item is Alt+F.

### **Load Map File**

Load a map file to be digitized. This option allows the user to navigate through the whole disk and select the desired file. File selected is loaded in the window for map digitization purpose.

### **Load Result File**

This option is useful to analyze the old results. User can not modify or change the result file (i.e., a map), it can only navigate through different output files created whenever a problem is solved.

### **Exit**

This option terminates the program execution.

## **The Mark Menu**

This menu has the following options.

### **Mark Outer**

To digitize a map, click on the outer boundary of the map. When user clicks on the Mark Outer menu option, system displays a check mark on it, this is the indication of system being ready for digitization. If whole boundary has been clicked on a number of points, and the user wants to fill up the inner details of a map e.g., cities, the he has to again click the mark outer menu. System removes the check mark and becomes ready for next operation.

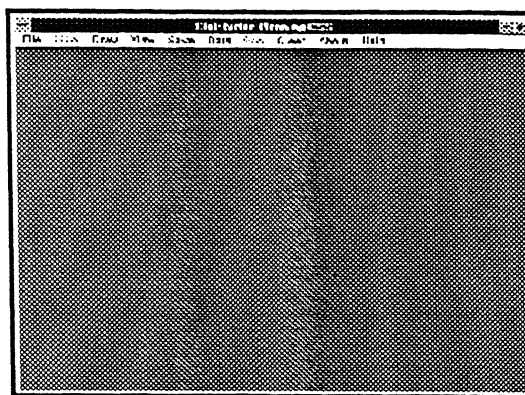


Fig. 6.1 Basic Layout of screen

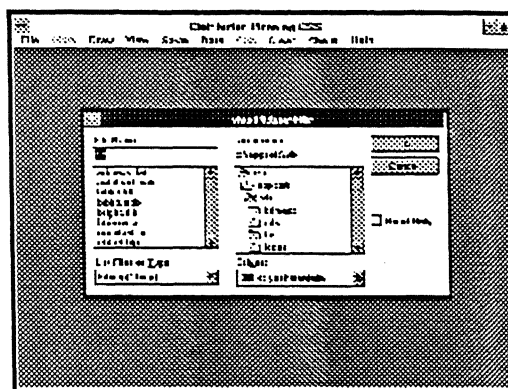


Fig. 6.2 Basic dialog box structure

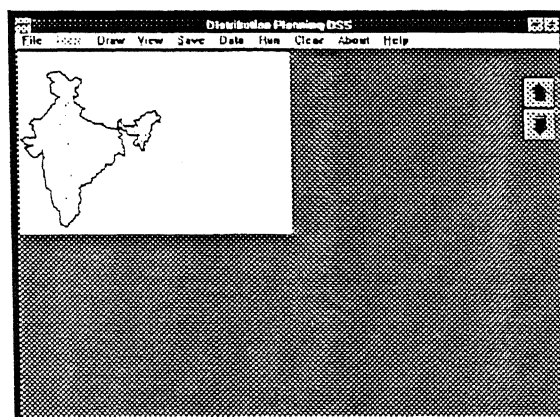


Fig 6.3 Digitized map displayed on screen

### **Mark Inner**

This makes the system ready to store the city name as well as the location of city on the map.

### **The Draw Menu**

This option can be used to draw a map on the screen. Various option are as follows.

#### **Draw Directly**

This option is provided to draw a map on the screen which has recently been digitized. This option does not store the point locations in a file.

#### **By Draw File**

This option is also used to draw a map on the screen. It take input from a file. Map drawn using this option can be scaled up or down. Separate buttons are provided for this.

### **The View Menu**

This option is used to control the size of the map on the screen. Sub-options of this menu are like..

#### **Full Page**

Any map drawn by the methods given in this application can be enlarged to fit in the entire screen.

#### **Zoom In/Out**

This option can be used to increase or decrease the map size in steps of 10%.

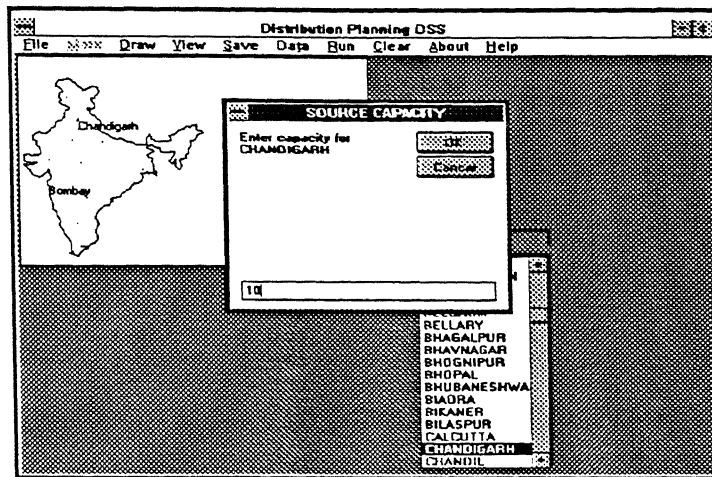


Fig. 6.4 Screen showing selection of cities from a list.

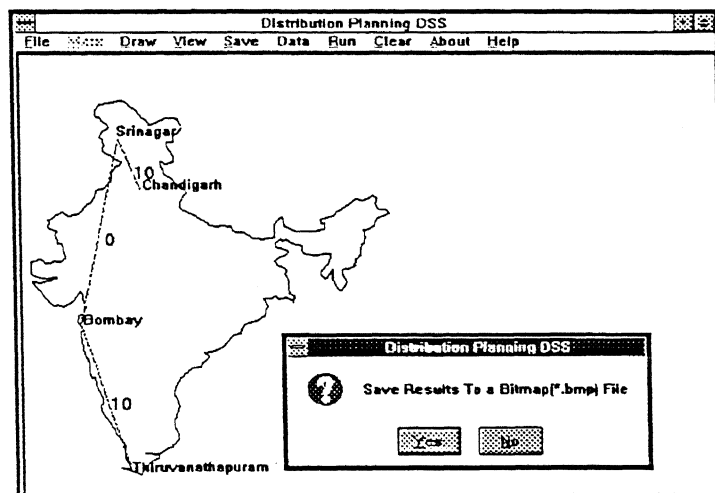


Fig. 6.5 Final output displayed on user screen.

## **The Save Menu**

This option has only one sub-option.

### **Save Map File**

This option is used to store the location values of different points clicked during the digitization process. If no point is clicked and option is selected, the system gives an error message.

## **The Data Menu**

This menu option is a two level menu. It has the following options.

### **Choose Sources / Destination**

This option asks the user to provide the source/destination data file name and automatically adds the **.dat** extension.

### **Start Selection**

Choose this option to select a set of cities from a list provided by the system. As the user clicks on a particular city given in the list, the system asks to give capacity/requirement of that particular source/destination.

### **End Selection**

To close the source data file opened to store the data. All the information given by the user is cleared from the buffer and dumped into the file.

### **Distance Table**

This option is used only when the user wants to create a new distance database or wants to append the existing database.

### **Prepare Cost Table**

This option is used to generate the basic input data file. This option asks the user to supply the file name where the data will be stored. Input to this file is taken from the source data file, destination data file, and the distance table.

### **The Run Menu**

#### **Execute Program**

Choose this option to solve the transportation problem. Option asks the user to name the file where the solution is going to be stored. This file becomes the basic input to display the map representation of the solution.

#### **Show Results**

This option presents a map with all the sources and destinations represented on it with different lines showing the quantity transferred from a particular source to a particular destination. Quantity transferred is displayed at the middle of line.

### **The Clear Menu**

This option is chosen whenever the image drawn in the window area needs to be wiped out. Any information on the screen is cleared and screen becomes blank.

### **The About Menu**

This option displays a window, which tells the user about the license and copyright information.

### **The Help Menu**

To invoke help, choose this option.



## **The System's Error Messages**

The system generates error messages as soon as the user violates the logical flow of operations and gives the information in terms of the possible cause of error and the remedy.

### **Nothing To Draw**

This error comes whenever user selects draw directly option without having clicked any point. It asks the user to first click some points then try the option.

### **.dss File Not Loaded**

Whenever the user clicks the option for source or destination selection, the system simultaneously displays the location of the city selected. If in case user forgets and do not load the draw file on which the cities will be displayed, system generates an error and asks the user to load the map file on which the cities will be displayed.

### **Source/Destination File Missing**

When the user tries to create a cost matrix file without loading necessary file(s) e.g., source or destination file(s), system generates an error and asks the user to supply the needed file(s).

### **Can't Execute**

When system tries to run the transportation program and one or more file(s) needed (source, destination or cost matrix) is missing, system generates an error message.

## BIBLIOGRAPHY

1. Bradley, S.P., A.C. Hax, and T.L. Magnanti, Applied Mathematical Programming, Addison-Wesley, Reading, Mass., 1977.
2. Burrough, P.A., Principles of Geographical Information System for Land Resources Assessment, Claredon Press, Oxford, England, 1986.
3. Business Computer Systems, July 1984.
4. Daellenbach, H.G., and E.J. Bell, User's Guide to Linear Programming, Prentice Hall, Englewood Cliffs, N.J., 1970.
5. Driebeek, N.J., Applied Linear Programming, Addison-Wesley, Reading, Mass., 1969.
6. Fabozzi, E.J., and J. Valente, Mathematical Programming in American Companies: A Sample Survey, Interfaces, 7(1), November 1976, pp. 93-98.
7. Gass, S.I., An Illustrated Guide to Linear Programming, McGraw-Hill, New York, 1970.
8. Glover, F., D. Karney, and D. Klingman, The Augment Predecessor Index Method for Locating Distribution Problems, Transportation Science, 6, 1, pp. 171-180, 1972.
9. Gray, P., and C. Cullinan-James, Applied Optimization-A Survey, Interfaces, 6(3), May 1976, pp. 24-41.
10. Greenwood, A.J., and L.J. Moore, A Computer Based Decision Support System for University Tuition and Fee Policy, Joint National Meeting of TIMS/ORSA, Boston, MA, April 1985.
11. Gupta, R.K., and P. Lenin, Computer Aided Transportation and Distribution System for Fertilizer Movement, 26th Annual Convention of the Operational Research Society of India, December 6-8, 1993.
12. Hillier, F.S., and Gerald J. Lieberman, Introduction to Operations Research, 3rd Edition, Holden-Day, San Francisco, California, 1980.
13. Johnson, E., Networks and Basic Solutions, Operations Research, 14, 4, pp. 89-95, 1966.
14. Keen, P.G.W., and M.S.Scott-Morton, Decision Support System, An Organizational Perspective, Reading, MA: Addison Wesley, 1978.

15. Keen, P.G.W., Decision Support Systems Translating Analytic Techniques into Useful Tools, Sloan Management Review, Vol. 21, spring 1989, pp.. 33-44.
16. Medsker, L.R., An Interactive DSS for Energy Policy Analysis, Communication of the ACM, Nov., 1984.
17. Murty, K.G., Linear Programming, Wiley, New York, 1983.
18. Phillips, Don T., Ravindran, A., Soleberg, James J., Operations Research: Problems and Practice, John Wiley, 1976.
19. Reimann, B.C., Decision Support Systems: Strategic Management Tools for Eighties, Business Horizons, 1985
20. Salkin, H.M., and J. Saha, Studies in Linear Programming, North Holland/American Elsevier, New York, 1975.
21. Sarma, G.V., The Reduced Matrix Method to Solve the Transportation Problem, Opsearch, Vol. 31, No. 1, 1994.
22. Scott-Morton, M.S., Management Decision System: Computer Based Support for Decision Making, Cambridge, MA: Division of Research, Harvard University, 1971.
23. Steen, L.A., Linear Programming: Solid New Algorithm, Science News, Oct. 6, 1979, pp. 116, 234-236.
24. Taha, Hamdy A., Operations Research An Introduction, 2nd Edition, Macmillan, New York, 1976.
25. Turban, E., Decision Support and Expert Systems: Management Support System, 2nd Edition, Maxwell Macmillan International Editions, New York, 1992.

119129

## Date Slip

119129

This image shows a blank sheet of white paper with horizontal ruling lines. A solid vertical line runs down the center of the page, creating two equal-width columns. The horizontal lines are evenly spaced and extend across the entire width of the page, including both columns. There are no markings, text, or other graphical elements on the paper.

IME-1995-M-GDP-MAP